

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE CMSUBR

OBJECT MODULE PLACED IN CMSUBR.OBJ

COMPILER INVOKED BY: :F0: CMSUBR.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982) XREF

```

1      $title ('CODEHACRO SUBROUTINE MODULE')
      cmsubr:
      do;

```

```

/*

```

```

      modified 4/7/81 R. Silberstein
      modified 4/13/81 R. Silberstein
      modified 5/5/81 R. Silberstein
      modified 7/2/81 R. Silberstein

```

```

*/

```

```

/*

```

```

      This is the module to

```

```

      1) test if a set of operands matches a given instruction

```

```

      and

```

```

      2) produce output code for matched instruction

```

```

      The module interfaces the CODEOUTPUT module to
      physically send code bytes to the HEX output file.

```

```

*/

```

```

      $include (:f1:macro.lit)
= $nolist
      $include (:f1:equals.lit)
= $nolist
      $include (:f1:caacd.lit)
= $nolist
      $include (:f1:outp.lit)
= $nolist
      $include (:f1:scan.ext)
= $nolist
      $include (:f1:subr1.ext)
= $nolist
      $INCLUDE (:F1:SUBR2.EXT)
= $nolist
      $include (:f1:outp.ext)
= $nolist
      $include (:f1:ermmod.ext)
= $nolist
      $include (:f1:cmsubr.x86)
= $nolist

```

CR/m-86 v.1.1 (vol. IV)
 COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # C81-162

```
90 1      %object
      dcl          /* global variables */
      bytevar based macroptr byte, /* variables within codemacros */
      addrvar based macroptr addr,
      emitbyte(80) byte,          /* buffer of output codebytes */
      emitindex   byte,          /* index of 'emitbyte' */
      bitcomtab(2) byte data /* legal commands within 'DBIT' */
      (mnumberbits,mformalbits);
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$eject
```

```
/***** MICHELLANEDUS SUBROUTINES: *****/
```

```

91 1  clearcmindex: proc public;
92 2      emitindex=0;
93 2  end clearcmindex;

94 1  emit: proc public; /* emit codebytes for an instruction */
95 2      dcl i byte;
96 2      i=0ffh;
97 2      do while (i:=i+1) < emitindex;
98 3          call emitcodebyte(emitbyte(i),CSdata);
99 3      end$while;
100 2  end emit;

101 1  emitdummies: proc public; /* emit dummy (NO-OP-) bytes if error */
102 2      dcl (i,j) byte,nodum(4) byte data(2,5,6,8);
103 2      j=nooper;
104 2      if j>3 then j=3;
106 2      i=0ffh;
107 2      do while (i:=i+1) < nodum(j);
108 3          call emitcodebyte(90h,CSdata); /* 90H = NOP */
109 3      end$while;
110 2  end emitdummies;

111 1  emitsinglebyte: proc(ch); /* fill local emitbuffer with a new byte */
112 2      dcl ch byte;
113 2      if noerror then$do
115 3          emitbyte(emitindex)=ch;
116 3          emitindex=emitindex+1;
117 3      end$if;
118 2  end emitsinglebyte;

119 1  emitsingleword: proc (var); /* fill 2 new bytes into emitbuffer */
120 2      dcl var addr, byt1 byte at(.var), byt2 byte at(.var+1);
121 2      call emitsinglebyte(byt1);
122 2      call emitsinglebyte(byt2);
123 2  end emitsingleword;

124 1  incrmacroptr: proc;
125 2      macroptr=macroptr+1;
126 2  end incrmacroptr;

127 1  getoperadr: proc address;
128 2      dcl pt address;
129 2      pt=.operands(bytevar);
130 2      call incrmacroptr;
131 2      return pt;
132 2  end getoperadr;

/* recognize codemacro command type */
133 1  commandtype: proc(comno,lq,pt) byte public;
134 2      dcl (comno,lq,i) byte,pt address,ch based pt(1) byte;
135 2      i=0ffh;
136 2      do while (i:=i+1) < lq;
137 3          if comno=ch(i) then$do call incrmacroptr; return i; end$if;

```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 578
PACIFIC GROVE, CA 93950

SER. # _____

```
143 2        endwhile;  
144 2        return lg;  
      end commandtype;
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
%eject
```

```
/***** CODEMACRO COMMAND SUBROUTINES: *****/
```

```

145 1  mDBNROUT: proc public;
146 2      call emitsinglebyte(bytevar);
147 2      call incrmacroptr;
148 2  end mDBNROUT;

149 1  mDBFRout: proc public;
150 2      dcl pt address,opr based pt operandstruc;
151 2      pt=getoperadr;
152 2      call emitsinglebyte(opr.offset);
153 2  end mDBFRout;

154 1  mDWNROUT: proc public;
155 2      call emitsingleword(addrvar);
156 2      call incrmacroptr;
157 2      call incrmacroptr;
158 2  end mDWNROUT;

159 1  mDWFROUT: proc public;
160 2      dcl pt address,opr based pt operandstruc;
161 2      pt=getoperadr;
162 2      call emitsingleword(opr.offset);
163 2  end mDWFROUT;

164 1  mDDFROUT: proc public;
165 2      dcl pt address,opr based pt operandstruc;
166 2      pt=getoperadr;
167 2      if (opr.sflag and seqmbit) = 0 then call errmsg(missequinfo);
168 2      call emitsingleword(opr.offset);
169 2      call emitsingleword(opr.segment);
170 2      call emitsingleword(opr.segment);
171 2  end mDDFROUT;

172 1  mRELBROUT: proc public;
173 2      dcl pt address,opr based pt operandstruc,displ addr;
174 2      pt=getoperadr;
175 2      displ=opr.offset-cip-2;
176 2      if (opr.segment <> csegvalue) or (typecalc(displ)=wrd) then%do
177 3          call errmsg(laboutofrange);
178 3      end%if;
179 3      call emitsinglebyte(displ);
180 2      IF ABSADDR (0) = SPACE THEN%DO
181 3          CALL HEX2OUT (OPR.OFFSET, .ABSADDR);
182 3      END%IF;
183 3      CALL HEX2OUT (OPR.OFFSET, .ABSADDR);
184 3      END%IF;
185 2  end mRELBROUT;

186 1  mRELWROUT: proc public;
187 2      dcl pt address,opr based pt operandstruc;
188 2      pt=getoperadr;
189 2      if opr.segment <> csegvalue then call errmsg(laboutofrange);
190 2      call emitsingleword(opr.offset-cip-3);
191 2      IF ABSADDR (0) = SPACE THEN%DO
192 3          CALL HEX2OUT (OPR.OFFSET, .ABSADDR);
193 3      END%IF;
194 3      CALL HEX2OUT (OPR.OFFSET, .ABSADDR);
195 3      END%IF;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

196 2   end mRELWout;

197 1   mNOSEGFIXout: proc public;
198 2       dcl (segr,flag,segt) byte,pt address,opr based pt operandstruc;
199 2       segr=bytevar;
200 2       call incrmacroptr;
201 2       pt=getoperadr;
202 2       if (opr.baseindex and nooverridebit) = 0 then do
204 3           flag=opr.sflag;
205 3           segt=shr(flag,segtypecount) and 3;
206 3           noerror=(segt=segr);
207 3       end if;
208 2   end mNOSEGFIXout;

209 1   mSEGFIXout: proc public;
210 2       dcl pt address,opr based pt operandstruc,(segr,override,sflag) byte;
211 2       DSovertest: proc byte;
212 3           segr=shr(opr.baseindex,baseregcount) and 1;
213 3           return (((sflag and bregbit) <> 0) and (segr=1)); /* 1 = BP */
214 3       end DSovertest;

215 2       pt=getoperadr;
216 2       sflag=opr.sflag;
217 2       if (opr.baseindex and nooverridebit) = 0 then do
219 3           segr=shr(sflag,segtypecount) and 3;
220 3           do case segr;
221 4               do; override=true; segr=ESover; end; /* ES */
225 4               do; override=true; segr=CSover; end; /* CS */
229 4               do; override=not DSovertest; segr=SSover; end; /* SS */
233 4               do; override=DSovertest; segr=DSovery; end; /* DS */
237 4           end case;
238 3           if override then call emitsinglebyte(segr);
239 3       end if;
241 2   end mSEGFIXout;

242 1   MODRM: proc (regfield,pt);
243 2       dcl pt address,opr based pt operandstruc,
           (regfield,modfield,rmfield,dispflag,stype,sflag,segr) byte,
           BASEIND BYTE,
           offset addr,
           disphigh byte at(.offset),disphigh byte at(.offset+1);

244 2       disptype: proc byte;
245 3           if segr=rsc then return 2; /* disp always 2 for variable in CS */
247 3           if offset = 0 then return 0;
249 3           return typecalc(offset);
250 3       end disptype;

251 2       indextype: proc byte;
252 3           if (sflag and iregbit) <> 0 then do
254 4               if (sflag and bregbit) <> 0 then return 0;
256 4               return 1;
257 4           end if;
258 3           return 2;
259 3       end indextype;

260 2       offset=opr.offset; /* pick up operand attributes */

```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER # _____

```

261 2      stype=opr.stype;
262 2      sflag=opr.sflag;
263 2      segr=shr(sflag,segtypecount) and 3;
264 2      BASEIND = OPR.BASEINDEX AND (BASEREGBIT OR INDEXREGBIT);

265 2      if stype=reg then$do
267 3          rfield=offset;
268 3          modfield=11b;
269 3          dispflag=0;
270 3      else$do
272 3          if (sflag and (iregbit or bregbit)) = 0 then$do
274 4              rfield=110b;
275 4              modfield=0;
276 4              dispflag=2;
277 4          else$do
279 4              dispflag=disptype;    /* get no of DISP bytes */
280 4              modfield=dispflag;
281 4              do case indextype;

282 5                  /* both base- and index-reg */
283 5                  RFIELD = BASEIND AND (INDEXREGBIT OR BASEREGBIT);

284 5                  /* index reg only */
285 5                  RFIELD = 100b OR (BASEIND AND INDEXREGBIT);

286 5                  do;    /* base reg only */
287 6                      IF (BASEIND AND BASEREGBIT) > 0 THEN$DO
288 7                          rfield=110b;
289 7                          /* mod=00 and r/m=110b is a special case */
290 8                          if dispflag=0 then$do
291 8                              dispflag,modfield=1;
292 8                              end$if;
293 8                          else$do
294 7                              rfield=111b;
295 7                              end$if;
296 6                              end;
297 5                          endtcase;
298 4                              end$if;
299 3                              end$if;
300 2                      regfield=shl(regfield,3) and 30h;
301 2                      modfield=shl(modfield,6) and 0c0h;
302 2                      call emitsinglebyte(regfield or modfield or rfield);
303 2                      if dispflag > 0 then$do
304 3                          call emitsinglebyte(displow);
305 3                          if dispflag=2 then call emitsinglebyte(disphigh);
306 3                          end$if;
307 2                      end MODRM;

310 1      mMODRM1rout: proc public;
311 2          dcl regfield byte;
312 2          regfield=bytevar;
313 2          call incrmacroptr;
314 2          call MODRM(regfield,getoperadr);
315 2          end mMODRM1rout;

316 1      mMODRM2rout: proc public;
317 2          dcl regfield byte,pt address,opr based pt operandstruc;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

318 2    pt=getoperadr;
319 2    regfield=opr.offset;
320 2    call MODRM(regfield,getoperadr);
321 2    end mMODRM2rout;

322 1    mDBITrout: proc public;
323 2    dcl (result,crbit) byte,bittab(8) byte data(1,2,4,8,16,32,64,128);

324 2    join: proc(numb,nobit,noshift);
325 3    dcl (numb,nobit,noshift) byte;
326 3    if noshift > 0 then numb=shr(numb,noshift);
328 3    if nobit < 8 then numb=shl(numb,8-nobit);
330 3    do while (crbit <> 0ffh) and (nobit > 0);
331 4        if (numb and 80h) <> 0 then result=result or bittab(crbit);
333 4        crbit=crbit-1;
334 4        nobit=nobit-1;
335 4        numb=shl(numb,1);
336 4    end$while;
337 3    end join;

338 2    NUMBERBITSrout: proc;
339 3    dcl nobit byte;
340 3    nobit=bytevar;
341 3    call incrmacroptr;
342 3    call join(bytevar,nobit,0);
343 3    call incrmacroptr;
344 3    end NUMBERBITSrout;

345 2    FORMBITSrout: proc;
346 3    dcl (nobit,numb) byte,pt address,opr based pt operandstruc;
347 3    nobit=bytevar;
348 3    call incrmacroptr;
349 3    pt=getoperadr;
350 3    numb=opr.offset;
351 3    call join(numb,nobit,bytevar);
352 3    call incrmacroptr;
353 3    end FORMBITSrout;

354 2    result=0;
355 2    crbit=7; /* current bit position */
356 2    do while bytevar <> mendbit; /* do until ENDBIT command */
357 3    do case commandtype(bytevar,length(bitcomtab),.bitcomtab);
358 4        call NUMBERBITSrout;
359 4        call FORMBITSrout;
360 4    do; end;
362 4    end$case;
363 3    end$while;
364 2    call incrmacroptr; /* skip ENDBIT command */
365 2    call emit$singlebyte(result);
366 2    end mDBITrout;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

subject

/***** ROUTINES TO MATCH OPERANDS TO INSTRUCTION *****/

```

/* test user operand against codemacro parameter */
367 1 matchsingleop: proc(opno) byte;
368 2   dcl (match,specletter,modletter,range,rangetype) byte,
      (rangev1,rangev2,opno) byte,
      pt address, oper based pt operandstruc;

369 2   rangetest: proc byte; /* perform rangetest */
370 3     dcl opvalue byte;
371 3     rangev1=bytevar;
372 3     call incrmacroptr;
373 3     if range=doublerange then$do
375 4       rangev2=bytevar;
376 4       call incrmacroptr;
377 4     end$if;
378 3     opvalue=oper.offset;
379 3     if range=doublerange then$do
381 4       return ((opvalue>=rangev1) and (opvalue<=rangev2));
382 4     else$do
384 4       return (opvalue=rangev1);
385 4     end$if;
386 3   end rangetest;

387 2   modlettertest: proc byte;
388 3     dcl numb addr,(styp,modbyt) byte;
389 3     styp=oper.stype;
390 3     if styp=lab then return true;
392 3     modbyt=oper.sflag and typebit;
393 3     if styp = reg then return (modbyt=modletter);
395 3     if styp = variable then
396 3       return ((modbyt=nomod) or (modbyt=modletter));
397 3     if styp=number then$do
399 4       numb=oper.offset;
400 4       do case modletter-1;
401 5         return not wrdtest(numb); /* BYTE */
402 5         return wrdtest(numb); /* WORD */
403 5         return (typecalc(numb)=byt); /* signed BYTE */
404 5         return false; /* DWORD */
405 5       end$case;
406 4     end$if;
407 3     return false;
408 3   end modlettertest;

409 2   speclettertest: proc byte;
410 3     dcl (opertype,locvalue,loctype) byte;
411 3     memtest: proc byte;
412 4       return (opertype=variable);
413 4     end memtest;

414 3     opertype=oper.stype;
415 3     locvalue=oper.offset;
416 3     loctype=oper.sflag and typebit;
417 3     do case specletter;

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

/* A - accumulator (AX or AL) */
118 4 return ((opertype=reg) and (locvalue=rax));
/* C - code reference, i.e. label */
119 4 return (opertype=label);
/* D - immediate data */
120 4 return (opertype=number);
/* E - effective address, i.e. memory address or register */
121 4 return (memtest or (opertype=reg));
/* M - memory address */
122 4 return memtest;
/* R - register except segment register */
123 4 return ((opertype=reg) and (loctype <= dword));
/* S - segment register */
124 4 return ((opertype=reg) and (loctype = dword));
/* X - memory address without indexing */
125 4 return ((opertype=variable) and
          ((oper.sflag and (iregbit or bregbit))=0));
126 4 end$case;
127 3 end speclettertest;

128 2 specletter=bytevar; /* pick up codemacro attributes */
129 2 call incrmacroptr;
130 2 modletter=bytevar and modletter$bit;
131 2 range=bytevar and range$and;
132 2 rangetype=bytevar and rangetype$and;
133 2 call incrmacroptr;
134 2 pt=.operands(opno); /* address of current user operand */

135 2 match=true;
136 2 if range <= norange then match=rangetest;
138 2 if modletter <= 0 then match=match and modlettertest;
140 2 if match then match=speclettertest;
142 2 return match;
143 2 end matchsingleop;

/* test if operands match a specific codemacro */
144 1 matchingops: proc byte;
145 2 dcl savept address, (nopara, match, parno) byte;
146 2 savept=macroptr;
147 2 call incrmacroptr; /* macroptr=macroptr+2 */
148 2 call incrmacroptr;
149 2 nopara=bytevar; /* pick up no of parameters */
150 2 call incrmacroptr; /* advance to first formal */
151 2 if (nopara and prefix$on) <= 0 then return true; /* PREFIX */
153 2 if nopara <= nooper then$do
154 3 match=false;
156 3 else$do
158 3 match=true;
159 3 parno=0fffh;
160 3 do while (parno:=parno+1) <= nopara;
161 4 match=match and matchsingleop(parno);
162 4 end$while;
163 3 end$if;
164 2 if not match then macroptr=savept;
166 2 return match;
167 2 end matchingops;

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

PL/M-86 COMPILER CODEMACRO SUBROUTINE MODULE

```
468 1      /* test if operands match instruction */
469 2      searchformatch: proc byte public;
470 2          dcl next based macroptr address;
471 2          macroptr=firstmacroptr;
472 3          do forever;
473 3              if matchingops then return true;
474 3              if next=0 then return false;
475 3              macroptr=next;
476 3          end$forever;
477 2      end searchformatch;
478 2
479 1      end$module casubr;
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

89	0000H	4	ABSADDR.	BYTE ARRAY(4) EXTERNAL(25)	181	183	192	194												
2			ADDR	LITERALLY	32	35	38	41	47	65	68	71	74	89	90					
					120	150	160	165	173	187	198	210	243	317	346	368	388			
90	0000H	2	ADDRVAR.	WORD BASED(MACROPTR)	155															
55	0000H		ALPHANUMERIC . . .	PROCEDURE BYTE EXTERNAL(14) STACK=0000H																
58	0000H		ASCIICHAR.	PROCEDURE BYTE EXTERNAL(15) STACK=0000H																
243	007FH	1	BASEIND.	BYTE	264	282	283	285												
160	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
193	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)	202															
346	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
165	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
173	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
210	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)	212	217														
89	0008H	1	BASEINDEX.	BYTE MEMBER(OPERANDS)																
107	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
150	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
317	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
243	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)	264															
363	0008H	1	BASEINDEX.	BYTE MEMBER(OPR)																
88			BASEREBIT	LITERALLY	264	282	285													
88			BASEREGCOUNT . . .	LITERALLY	212															
90	0000H	2	BITCOMTAB.	BYTE ARRAY(2) DATA	357															
323	0003H	8	BITTAB	BYTE ARRAY(8) DATA	332															
88			BREGBIT.	LITERALLY	213	254	272	425												
88			BREGCOUNT.	LITERALLY																
4			BYT.	LITERALLY	403															
120	0004H	1	BYT1	BYTE AT AUTOMATIC	121															
120	0005H	1	BYT2	BYTE AT AUTOMATIC	122															
90	0000H	1	BYTEVAR.	BYTE BASED(MACROPTR)	129	146	199	312	340	342	347	351	356							
					357	371	375	428	430	431	432	449								
111	0004H	1	CH	BYTE PARAMETER AUTOMATIC	112	115														
80	0000H	1	CH	BYTE PARAMETER	81															
49	0000H	1	CH	BYTE PARAMETER	50															
52	0000H	1	CH	BYTE PARAMETER	53															
58	0000H	1	CH	BYTE PARAMETER	59															
55	0000H	1	CH	BYTE PARAMETER	56															
61	0000H	1	CH	BYTE PARAMETER	62															
134	0000H	1	CH	BYTE BASED(PT) ARRAY(1)	137															
46	0000H	1	CH	BYTE PARAMETER	47															
89	0000H	2	CIP.	WORD EXTERNAL(26)	175	191														
91	0000H	10	CLEARCINDEX . . .	PROCEDURE PUBLIC STACK=0002H																
1	0000H		CMSUBR	PROCEDURE STACK=0000H																
3			CODE	LITERALLY																
133	00C3H	50	COMMANDTYPE. . . .	PROCEDURE BYTE PUBLIC STACK=000CH	357															
133	0008H	1	COMNO.	BYTE PARAMETER AUTOMATIC	134	137														
37	0000H		COPY	PROCEDURE EXTERNAL(8) STACK=0000H																
2			CR	LITERALLY																
323	0083H	1	CRBIT.	BYTE	330	332	333	355												
16			CSDATA	LITERALLY	98	108														

DEPARTMENT OF AGRICULTURE
 DIGITAL INFORMATION
 9000 E. 17th
 DENVER, CO 80202
 SER. # _____

89	0000H	2	CSEGVVALUE.	WORD EXTERNAL(27)	176	139
11			CSOVER	LITERALLY	227	
16			CSVALUE.	LITERALLY		
37	0000H	2	D.	WORD PARAMETER	38	
73	0000H	2	D.	WORD PARAMETER	74	
70	0000H	2	D.	WORD PARAMETER	71	
67	0000H	2	D.	WORD PARAMETER	68	
64	0000H	2	D.	WORD PARAMETER	65	
			DCL.	LITERALLY		
73	0000H		DECOUT	PROCEDURE EXTERNAL(20) STACK=0000H		
3			DELETEDSYMB.	LITERALLY		
49	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(12) STACK=0000H		
243	007BH	1	DISPFLAG	BYTE	269 276 279 280 288 290 303 306	
243	0013H	1	DISPHIGH	BYTE AT	307	
177	000AH	2	DISPL.	WORD	175 176 180	
243	0012H	1	DISPLOW.	BYTE AT	305	
244	03F0H	34	DISPTYPE	PROCEDURE BYTE STACK=0006H		
3			DOUBLEDEFINED.	LITERALLY		
15			DOUBLERANGE.	LITERALLY	373 379	
16			DRCODEDATA	LITERALLY		
16			DRCODESEGM	LITERALLY		
16			DRDATADATA	LITERALLY		
16			DRDATASEGM	LITERALLY		
16			DREXTRADATA.	LITERALLY		
16			DREXTRASEGM.	LITERALLY		
16			DRSTACKDATA.	LITERALLY		
16			DRSTACKSEGM.	LITERALLY		
16			DSDATA	LITERALLY		
11			DSOVER	LITERALLY	235	
211	02CDH	45	DSOVERTEST	PROCEDURE BYTE STACK=0004H	230 234	
16			DSVALUE.	LITERALLY		
4			DWRD	LITERALLY	423 424	
2			ELSEDO	LITERALLY	291	
94	000AH	40	EMIT	PROCEDURE PUBLIC STACK=0008H		
90	001EH	80	EMITBYTE	BYTE ARRAY(30)	98 115	
80	0000H		EMITCODEBYTE	PROCEDURE EXTERNAL(23) STACK=0000H	98 108	
101	0032H	56	EMITDUMMIES.	PROCEDURE PUBLIC STACK=0008H		
90	006EH	1	EMITINDEX.	BYTE	92 97 115 116	
76	0000H		EMITINIT	PROCEDURE EXTERNAL(21) STACK=0000H		
111	006AH	31	EMITSINGLEBYTE	PROCEDURE STACK=0004H	121 122 146 152 180 239 302 305	
					307 335	
119	0089H	19	EMITSINGLEWORD	PROCEDURE STACK=000AH	155 162 169 170 191	
73	0000H		EMITTERMINATE.	PROCEDURE EXTERNAL(22) STACK=0000H		
29	0000H		EMPTYLINE.	PROCEDURE BYTE EXTERNAL(5) STACK=0000H		
2			ENDCASE.	LITERALLY		
2			ENDDO.	LITERALLY		
2			ENDFOREVER	LITERALLY		
2			ENDIF.	LITERALLY	239 307	
2			ENDMODULE.	LITERALLY		
2			ENDOFFILE.	LITERALLY		
2			ENDPROC.	LITERALLY		
2			ENDWHILE	LITERALLY	141	
16			EOFTYPE.	LITERALLY		
64	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(17) STACK=0000H		
83	0000H		ERRMSG	PROCEDURE EXTERNAL(24) STACK=0000H	168 178 190	
83	0000H	1	ERRNO.	BYTE PARAMETER	84	
3			ERROR.	LITERALLY		

277

COPYRIGHT © 1981, 1982

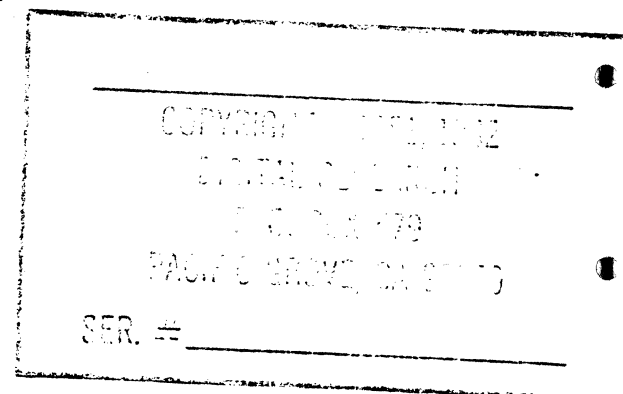
DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER #

Address	Hex	Symbol	Value	Comment	Address	Hex	Symbol	Value	Comment
16		ESDATA	LITERALLY						
11		ESOVER	LITERALLY	223					
16		ESVALUE	LITERALLY						
2		FALSE	LITERALLY	404 407 455 475					
43	0000H	2 FCBADR	WORD PARAMETER	44					
43	0000H	FILEABORT	PROCEDURE EXTERNAL(10) STACK=0000H						
46	0000H	FILL	PROCEDURE EXTERNAL(11) STACK=0000H						
89	0000H	2 FIRSTMACROPTR	WORD EXTERNAL(29)	470					
198	0074H	1 FLAG	BYTE	204 205					
2		FOREVER	LITERALLY	471					
345	0536H	53 FORMBITSROUT	PROCEDURE STACK=000EH	359					
2		FORMFEED	LITERALLY						
89	0000H	1 FULLSYHBTAB	BYTE EXTERNAL(31)						
127	00A5H	30 GETOPERADR	PROCEDURE WORD STACK=0006H	151 161 166 174 188 201 215					
			314 318 320 349						
67	0000H	HEX1OUT	PROCEDURE EXTERNAL(18) STACK=0000H						
70	0000H	HEX2OUT	PROCEDURE EXTERNAL(19) STACK=0000H	183 194					
134	0072H	1 I	BYTE	135 136 137 140					
102	0070H	1 I	BYTE	106 107					
95	006FH	1 I	BYTE	96 97 98					
3		IDENT	LITERALLY						
124	009CH	9 INCHMACROPTR	PROCEDURE STACK=0002H	130 139 147 156 157 200 313 341					
			343 348 352 364 372 376 429 433 447 448 450						
88		INDEXREGBIT	LITERALLY	264 282 283					
88		INDEXREGCOUNT	LITERALLY						
251	0412H	29 INDEXTYPE	PROCEDURE BYTE STACK=0002H	281					
2		INIT	LITERALLY						
16		INTELDATA	LITERALLY						
16		INTELSEGMENT	LITERALLY						
88		IREGBIT	LITERALLY	252 272 425					
88		IREGCOUNT	LITERALLY						
102	0071H	1 J	BYTE	103 104 105 107					
324	04B6H	72 JOIN	PROCEDURE STACK=000AH	342 351					
3		LAB	LITERALLY	390 419					
86		LABOUTOFRANGE	LITERALLY	178 190					
7		LEFTBRACKET	LITERALLY						
150	0000H	2 LENGTH	WORD MEMBER(OPR)						
346	0000H	2 LENGTH	WORD MEMBER(OPR)						
160	0000H	2 LENGTH	WORD MEMBER(OPR)						
210	0000H	2 LENGTH	WORD MEMBER(OPR)						
165	0000H	2 LENGTH	WORD MEMBER(OPR)						
317	0000H	2 LENGTH	WORD MEMBER(OPR)						
187	0000H	2 LENGTH	WORD MEMBER(OPR)						
173	0000H	2 LENGTH	WORD MEMBER(OPR)						
198	0000H	2 LENGTH	WORD MEMBER(OPR)						
368	0000H	2 LENGTH	WORD MEMBER(OPR)						
243	0000H	2 LENGTH	WORD MEMBER(OPR)						
		LENGTH	BUILTIN	357					
89	0000H								



				347	351	356	357	371	375	428	430	431	432	446	449	465	469
				470	474	476											
368	0087H	1	HATCH.	BYTE		435	437	439	440	441	442						
445	0095H	1	HATCH.	BYTE		455	458	461	464	466							
444	07A2H	108	MATCHINGOPS. . . .	PROCEDURE	BYTE	STACK=0010H					472						
367	056BH	111	MATCHSINGLEOP. . .	PROCEDURE	BYTE	STACK=000CH					461						
8			MDBF.	LITERALLY													
149	0106H	21	MDBFROUT.	PROCEDURE	PUBLIC	STACK=000AH											
8			MDBIT.	LITERALLY													
322	046BH	75	MDBITROUT.	PROCEDURE	PUBLIC	STACK=0012H											
8			MDBN.	LITERALLY													
145	00F5H	17	MDBNROUT.	PROCEDURE	PUBLIC	STACK=0008H											
8			MDDF.	LITERALLY													
164	0144H	47	MDDFROUT.	PROCEDURE	PUBLIC	STACK=000EH											
8			MDF.	LITERALLY													
159	012FH	21	MDFROUT.	PROCEDURE	PUBLIC	STACK=000EH											
8			MDWN.	LITERALLY													
154	011BH	20	MDWNROUT.	PROCEDURE	PUBLIC	STACK=000EH											
411	077FH	15	MEMENTEST.	PROCEDURE	BYTE	STACK=0002H					421	422					
8			MENDBIT.	LITERALLY		356											
8			MENDM.	LITERALLY													
8			MFORMALBITS. . . .	LITERALLY		90											
86			MISSEGINF.	LITERALLY		168											
8			MMODRM1.	LITERALLY													
310	042FH	28	MMODRM1ROUT. . . .	PROCEDURE	PUBLIC	STACK=0012H											
8			MMODRM2.	LITERALLY													
316	044BH	32	MMODRM2ROUT. . . .	PROCEDURE	PUBLIC	STACK=0012H											
8			MNOSEGFIX.	LITERALLY													
197	0218H	60	MNOSEGFIXROUT. . .	PROCEDURE	PUBLIC	STACK=000AH											
8			MNUMBERBITS. . . .	LITERALLY		90											
10			MODB.	LITERALLY													
388	0090H	1	MODBYT.	BYTE		392	394	396									
10			MODD.	LITERALLY													
243	0079H	1	MODFIELD.	BYTE		268	275	280	290	301	302						
368	0089H	1	MODLETTER.	BYTE		394	396	400	430	438							
14			MODLETTERBIT. . . .	LITERALLY		430											
14			MODLETTERCOUNT. .	LITERALLY													
387	063AH	162	MODLETTERTEST. . .	PROCEDURE	BYTE	STACK=0006H					439						
242	02F4H	246	MODRM.	PROCEDURE	STACK=000EH					314	320						

339	0084H	1	NOBIT.	BYTE	340	342													
346	0085H	1	NOBIT.	BYTE	347	351													
324	0006H	1	NOBIT.	BYTE PARAMETER AUTOMATIC			325	328	329	330	334								
102	0002H	4	NODUH.	BYTE ARRAY(4) DATA		107													
89	0000H	1	NOERROR.	BYTE EXTERNAL(28)		113	206												
10			NOMOD.	LITERALLY		396													
89	0000H	1	NOOPER.	BYTE EXTERNAL(32)		103	453												
88			NOOVERCOUNT. . . .	LITERALLY															
88			NOOVERRIDEBIT. . .	LITERALLY		202	217												
445	0094H	1	NOPARA.	BYTE	449	451	453	460											
12			NOPARAND.	LITERALLY															
15			NORANGE.	LITERALLY		436													
324	0004H	1	NOSHIFT.	BYTE PARAMETER AUTOMATIC				325	326	327									
346	0086H	1	NUMB.	BYTE	350	351													
324	0008H	1	NUMB.	BYTE PARAMETER AUTOMATIC				325	327	329	331	335							
388	001AH	2	NUMB.	WORD	399	401	402	403											
3			NUMBER.	LITERALLY		397	420												
338	0512H	36	NUMBERBITSROUT. .	PROCEDURE STACK=000EH				358											
15			NUMBERRANGE. . . .	LITERALLY															
7			OAND.	LITERALLY															
7			OEO.	LITERALLY															
346	0006H	2	OFFSET.	WORD MEMBER(OPR)				350											
317	0006H	2	OFFSET.	WORD MEMBER(OPR)				319											
243	0006H	2	OFFSET.	WORD MEMBER(OPR)				260											
198	0006H	2	OFFSET.	WORD MEMBER(OPR)															
187	0006H	2	OFFSET.	WORD MEMBER(OPR)				191	194										
173	0006H	2	OFFSET.	WORD MEMBER(OPR)				175	183										
165	0006H	2	OFFSET.	WORD MEMBER(OPR)				169											
160	0006H	2	OFFSET.	WORD MEMBER(OPR)				162											
150	0006H	2	OFFSET.	WORD MEMBER(OPR)				152											
210	0006H	2	OFFSET.	WORD MEMBER(OPR)															
89	0006H	2	OFFSET.	WORD MEMBER(OPERANDS)															
243	0012H	2	OFFSET.	WORD	243	247	249	260	267										
368	0006H	2	OFFSET.	WORD MEMBER(OPR)				378	399	415									
7			OGE.	LITERALLY															
7			OGT.	LITERALLY															
7			OLAST.	LITERALLY															
7			OLE.	LITERALLY															
7			OLENGTH.	LITERALLY															
7			OLT.	LITERALLY															
7			OMOD.	LITERALLY															
7			ONE.	LITERALLY															
7			ONOT.	LITERALLY															
7			OOFFSET.	LITERALLY															
7			OOR.	LITERALLY															
368	0000H	9	OPER.	STRUCTURE BASED(PT)		378	389	392	399	414	415	416	425						
87	0000H	36	OPERANDS.	STRUCTURE ARRAY(4) EXTERNAL(33)						129	434								
87			OPERANDSTRUC. . .	LITERALLY		89	150	160	165	173	187	198	210	243	317	346			
						368													
3			OPERATOR.	LITERALLY															
410	0091H	1	OPERTYPE.	BYTE	412	414	418	419	420	421	423	424	425						
370	008EH	1	OPERVALUE.	BYTE	378	381	384												
367	0004H	1	OPNO.	BYTE PARAMETER AUTOMATIC						368	434								
346	0000H	9	OPR.	STRUCTURE BASED(PT)						350									
317	0000H	9	OPR.	STRUCTURE BASED(PT)						319									
210	0000H	9	OPR.	STRUCTURE BASED(PT)						212	216	217							
198	0000H	9	OPR.	STRUCTURE BASED(PT)						202	204								

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P.O. BOX 679

PACIFIC GROVE, CA 93950

SER. #

187	0000H	9	OPR.	STRUCTURE BASED(PT)	189	191	194
173	0000H	9	OPR.	STRUCTURE BASED(PT)	175	176	183
165	0000H	9	OPR.	STRUCTURE BASED(PT)	167	169	170
160	0000H	9	OPR.	STRUCTURE BASED(PT)	162		
150	0000H	9	OPR.	STRUCTURE BASED(PT)	152		
243	0000H	9	OPR.	STRUCTURE BASED(PT)	260	261	262 264

7			OPTR	LITERALLY			
7			OSEG	LITERALLY			
7			OSHL	LITERALLY			
7			OSHORT	LITERALLY			
7			OSHR	LITERALLY			
7			OTYPE.	LITERALLY			
40	0000H		OUTTEXT.	PROCEDURE EXTERNAL(9) STACK=0000H			
210	0077H	1	OVERRIDE	BYTE	222	226	230 234 238
7			OXOR	LITERALLY			
445	0096H	1	PARNO.	BYTE	459	460	461
6			PCODEMACRO	LITERALLY			
6			PCSEG.	LITERALLY			
6			PDB.	LITERALLY			
6			PDBIT.	LITERALLY			
6			PDD.	LITERALLY			
6			PDSEG.	LITERALLY			
6			PDW.	LITERALLY			
6			PEJECT	LITERALLY			
6			PEND	LITERALLY			
6			PENDIF	LITERALLY			
6			PENDM.	LITERALLY			
6			PEQU	LITERALLY			
6			PESEG.	LITERALLY			
6			PIF.	LITERALLY			
6			PIFLIST.	LITERALLY			
6			PINCLUDE	LITERALLY			
6			PLIST.	LITERALLY			
6			PMDRM	LITERALLY			
6			PNOIFLIST.	LITERALLY			
6			PNOLIST.	LITERALLY			
6			PNOSEGFIX.	LITERALLY			
6			PORG	LITERALLY			
6			PPAGESIZE.	LITERALLY			
6			PPAGEWIDTH	LITERALLY			
6			PRB.	LITERALLY			
12			PREFIXAND.	LITERALLY			
13			PREFIXON	LITERALLY	451		
6			PRELB.	LITERALLY			
6			PRELW.	LITERALLY			
2			PROC	LITERALLY			

			17	19	21	23	26	29	31	34	55	58	67
	70	73	76	78	80	83	91	94	101	111	119	124	127 133
	145	149	154	159	164	172	186	197	209	211	242	244	251 310
	316	322	324	338	345	367	369	387	409	411	444	448	

6			PRS.	LITERALLY			
6			PRW.	LITERALLY			
6			PSEGFIX.	LITERALLY			
3			PSEUDU	LITERALLY			
6			PSINFORM	LITERALLY			
6			PSSEG.	LITERALLY			
346	0016H	2	PT	WORD	346	349	350
317	0014H	2	PT	WORD	317	318	319

COPYRIGHT © 1981-1982

DIGITAL EQUIPMENT

P.O. BOX 1570

FACSIMILE 617-253-1100

SER. #

210	0010H	2	PT	WORD	210	212	215	216	217
198	000EH	2	PT	WORD	198	201	202	204	
187	000CH	2	PT	WORD	187	188	189	191	194
173	0008H	2	PT	WORD	173	174	175	176	183
165	0006H	2	PT	WORD	165	166	167	169	170
160	0004H	2	PT	WORD	160	161	162		
150	0002H	2	PT	WORD	150	151	152		
133	0004H	2	PT	WORD	PARAMETER	AUTOMATIC		134	137
128	0000H	2	PT	WORD	129	131			
242	0004H	2	PT	WORD	PARAMETER	AUTOMATIC		243	260 261 262 264
46	0000H	2	PT	WORD	PARAMETER	47			
368	0018H	2	PT	WORD	368	378	389	392 399	414 415 416 425 434
6			PTITLE	LITERALLY					
5			RAH.	LITERALLY					
5			RAL.	LITERALLY					
368	008AH	1	RANGE.	BYTE	373	379	431	436	
15			RANGEAND	LITERALLY	431				
14			RANGESPECBIT	LITERALLY					
14			RANGESPECCOUNT	LITERALLY					
369	05DAH	76	RANGETEST.	PROCEDURE	BYTE	STACK=0006H		437	
368	008BH	1	RANGETYPE.	BYTE	432				
15			RANGETYPEAND	LITERALLY	432				
368	008CH	1	RANGEV1.	BYTE	371	381	384		
368	008DH	1	RANGEV2.	BYTE	375	381			
5			RAX.	LITERALLY	418				
5			RBH.	LITERALLY					
5			RBL.	LITERALLY					
5			RBP.	LITERALLY					
5			RBX.	LITERALLY					
5			RCH.	LITERALLY					
5			RCL.	LITERALLY					
5			RCS.	LITERALLY	245				
5			RCX.	LITERALLY					
5			RDH.	LITERALLY					
5			RDI.	LITERALLY					
5			RDL.	LITERALLY					
5			RDS.	LITERALLY					
5			RDX.	LITERALLY					
2			REENT.	LITERALLY					
3			REG.	LITERALLY	265	393	418	421 423	424
317	0081H	1	REGFIELD	BYTE	319	320			
311	0080H	1	REGFIELD	BYTE	312	314			
242	0006H	1	REGFIELD	BYTE	PARAMETER	AUTOMATIC		243	300 302
15			REGISTERRANGE.	LITERALLY					
5			RES.	LITERALLY					
323	0082H	1	RESULT	BYTE	332	354	365		
7			RIGHTBRACKET	LITERALLY					
243	007AH	1	RMFIELD.	BYTE	267	274	282 283 287	294	302
5			RSI.	LITERALLY					
5			RSP.	LITERALLY					
5			RSS.	LITERALLY					
64	0000H	2	S.	WORD	PARAMETER	65			
37	0000H	2	S.	WORD	PARAMETER	38			
445	001CH	2	SAVEPT	WORD	446	465			
19	0000H		SCAN	PROCEDURE	EXTERNAL(1)	STACK=0000H			
17	0000H		SCANINIT	PROCEDURE	EXTERNAL(0)	STACK=0000H			
468	081AH	43	SEARCHFORMATCH	PROCEDURE	BYTE	PUBLIC	STACK=001AH		

COPYRIGHT © 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

88	SEGBIT	LITERALLY	167						
88	SEGCOUNT	LITERALLY							
368 0004H	2 SEGMENT	WORD MEMBER(OPR)							
346 0004H	2 SEGMENT	WORD MEMBER(OPR)							
317 0004H	2 SEGMENT	WORD MEMBER(OPR)							
243 0004H	2 SEGMENT	WORD MEMBER(OPR)							
210 0004H	2 SEGMENT	WORD MEMBER(OPR)							
178 0004H	2 SEGMENT	WORD MEMBER(OPR)							
187 0004H	2 SEGMENT	WORD MEMBER(OPR)							
173 0004H	2 SEGMENT	WORD MEMBER(OPR)							
165 0004H	2 SEGMENT	WORD MEMBER(OPR)							
160 0004H	2 SEGMENT	WORD MEMBER(OPR)							
150 0004H	2 SEGMENT	WORD MEMBER(OPR)							
89 0004H	2 SEGMENT	WORD MEMBER(OPERANDS)							
243 007EH	1 SEGR	BYTE	245	263					
210 0076H	1 SEGR	BYTE	212	213	219	220	223	227	231 235 239
198 0073H	1 SEGR	BYTE	199	206					
178 0075H	1 SEGT	BYTE	205	206					
88	SEGTYPEBIT	LITERALLY							
88	SEGTYPECOUNT	LITERALLY	205	219	263				
368 0003H	1 SFLAG	BYTE MEMBER(OPR)				392	416	425	
346 0003H	1 SFLAG	BYTE MEMBER(OPR)							
317 0003H	1 SFLAG	BYTE MEMBER(OPR)							
243 007DH	1 SFLAG	BYTE	252	254	262	263	272		
243 0003H	1 SFLAG	BYTE MEMBER(OPR)				262			
210 0078H	1 SFLAG	BYTE	213	216	219				
210 0003H	1 SFLAG	BYTE MEMBER(OPR)				216			
178 0003H	1 SFLAG	BYTE MEMBER(OPR)				204			
187 0003H	1 SFLAG	BYTE MEMBER(OPR)							
173 0003H	1 SFLAG	BYTE MEMBER(OPR)							
165 0003H	1 SFLAG	BYTE MEMBER(OPR)				167			
160 0003H	1 SFLAG	BYTE MEMBER(OPR)							
150 0003H	1 SFLAG	BYTE MEMBER(OPR)							
89 0003H	1 SFLAG	BYTE MEMBER(OPERANDS)							
	SHL	BUILTIN	300	301	329	335			
	SHR	BUILTIN	205	212	219	263	327		
15	SINGLERANGE	LITERALLY							
21 0000H	SKIPRESTOFLINE	PROCEDURE EXTERNAL(2) STACK=0000H							
26 0000H	SKIPUNTIL	PROCEDURE BYTE EXTERNAL(4) STACK=0000H							
2	SPACE	LITERALLY	181	192					
3	SPEC	LITERALLY							
9	SPECA	LITERALLY							
9	SPECC	LITERALLY							
9	SPECD	LITERALLY							
9	SPECE	LITERALLY							
23 0000H	SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(3) STACK=0000H							
368 0088H	1 SPECLETTER	BYTE	417	428					
409 06DCH	175 SPECLETTERTEST	PROCEDURE BYTE STACK=0006H						441	
9	SPECH	LITERALLY							
9	SPECR	LITERALLY							
9	SPECS	LITERALLY							
9	SPECX	LITERALLY							
16	SSDATA	LITERALLY							
11	SSOVER	LITERALLY	231						
16	SSVALUE	LITERALLY							
16	STARTTYPE	LITERALLY							
3	STRING	LITERALLY							

189

176

170

2	STRUC.	LITERALLY	89	150	160	165	173	187	198	210	243	317	346
		368											
388	008FH	1 STYP	BYTE	389	390	393	395	397					
368	0002H	1 STYPE.	BYTE MEMBER(OPR)				389	414					
346	0002H	1 STYPE.	BYTE MEMBER(OPR)										
317	0002H	1 STYPE.	BYTE MEMBER(OPR)										
243	007CH	1 STYPE.	BYTE	261	265								
243	0002H	1 STYPE.	BYTE MEMBER(OPR)				261						
210	0002H	1 STYPE.	BYTE MEMBER(OPR)										
198	0002H	1 STYPE.	BYTE MEMBER(OPR)										
187	0002H	1 STYPE.	BYTE MEMBER(OPR)										
173	0002H	1 STYPE.	BYTE MEMBER(OPR)										
165	0002H	1 STYPE.	BYTE MEMBER(OPR)										
160	0002H	1 STYPE.	BYTE MEMBER(OPR)										
150	0002H	1 STYPE.	BYTE MEMBER(OPR)										
89	0002H	1 STYPL.	BYTE MEMBER(OPERANDS)										
3	SYMBOL	LITERALLY											
87	SYMBOLSTRUC. . . .	LITERALLY	89	150	160	165	173	187	198	210	243	317	346
		368											
40	0000H	2 T.	WORD PARAMETER	41									
2	TAB.	LITERALLY											
43	0000H	2 TEXTADR.	WORD PARAMETER	44									
2	THENDO	LITERALLY	113	137	176	181	192	202	217	252	265	272	285
		288	303	373	379	397	453						
26	0000H	1 TOK.	BYTE PARAMETER	27									
23	0000H	1 TOK.	BYTE PARAMETER	24									
2	TRUE	LITERALLY	222	226	391	435	452	458	471	473			
80	0000H	1 TYP.	BYTE PARAMETER	81									
38	TYPEBIT.	LITERALLY	392	416									
31	0000H	TYPECALC	PROCEDURE BYTE EXTERNAL(6) STACK=0000H						176	249	403		
38	TYPECOUNT.	LITERALLY											
3	UDEFSYMB	LITERALLY											
61	0000H	UPPER.	PROCEDURE BYTE EXTERNAL(16) STACK=0000H										
31	0000H	2 VAL.	WORD PARAMETER	32									
119	0004H	2 VAR.	WORD PARAMETER AUTOMATIC				120						
3	VARIABLE	LITERALLY	395	412	425								
4	WRD.	LITERALLY	176										
34	0000H	WRDTEST.	PROCEDURE BYTE EXTERNAL(7) STACK=0000H						401	402			

MODULE INFORMATION:

CODE AREA SIZE = 0845H 2117D
 CONSTANT AREA SIZE = 000EH 14D
 VARIABLE AREA SIZE = 0097H 151D
 MAXIMUM STACK SIZE = 0014H 20D
 933 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____